Function Space Norms, and the Neural Tangent Kernel

Chendi Wang

March 16, 2021

A lecture based on Chapter 8 of Deep learning theory lecture notes

Outline

- Function class with infinitely many basis functions
- Function space norms
- Reproducing kernel Hilbert spaces
- Supervised machine learning
- Tangent models and neural tangent kernels

Function class with infinitely many basis functions

- Consider a measurable input space $\mathcal{X} \subseteq \mathbb{R}^d$ and a measurable parameter space $\mathcal{V} \subseteq \mathbb{R}^d$.
- Let $\{\varphi_v : \mathcal{X} \to \mathbb{R}\}_{v \in \mathcal{V}}$ be a set of continuous basis functions parametrized by $v \in \mathcal{V}$.
- For single-hidden-layer neural networks, one has $\varphi_v(x) = \sigma(w^\top x + b)$ with $\sigma : \mathbb{R} \to \mathbb{R}$ being an activation function. Here we denote $v = (w^\top, b)^\top$ with $w \in \mathbb{R}^{d-1}$ and $b \in \mathbb{R}$.
- Let τ be a probability measure on $\mathcal{V}$ and let $L^1(\tau)$ be the space of all integrable functions with respect to τ. We introduce a function space $\mathcal{F}_1$ by

$$\mathcal{F}_1 = \left\{ f : \mathcal{X} \to \mathbb{R} \,\middle|\, f(x) = \int_{\mathcal{V}} \varphi_v(x) p(v) d\tau(v), p \in L^1(\tau) \right\}.$$

Variation norm on $\mathcal{F}_1$

- For a given signed measure μ_p on $\mathcal{V}$ which has density $p \in L^1(\tau)$, the total variation of μ_p is given by

$$|\mu_p|(\mathcal{V}) := \int_{\mathcal{V}} |p(v)| d\tau(v) < +\infty.$$

- For any function $f \in \mathcal{F}_1$, the variation norm $\gamma_1(f)$ is the infimal value of $|\mu_p|(\mathcal{V})$ over all $p \in L^1(\tau)$ such that $f(x) = \int_{\mathcal{V}} p(v)\varphi_v(x) d\tau(v)$.
- For simplicity, we consider only the case with density functions. Note that not all measures have densities. One can generalize the corresponding theory to Radon measures (Kurkova and Sanguineti, 2001; Mhaskar, 2004; Bach, 2017).

Corresponding reproducing kernel Hilbert space

- Let $L^2(\tau)$ be the space of all square integrable functions w.r.t. τ. We now consider a new class of functions

$$\mathcal{F}_2 = \left\{ f : \mathcal{X} \to \mathbb{R} \middle| f(x) = \int_{\mathcal{V}} p(v)\varphi_v(x) d\tau(v), p \in L^2(\tau) \right\}.$$

- For $f \in \mathcal{F}_2$, we define a squared norm $\gamma_2^2(f)$ as the infimal value of $\int_{\mathcal{V}} |p(v)|^2 d\tau(v)$ over all p such that $f(x) = \int_{\mathcal{V}} p(v)\varphi_v(x) d\tau(v)$.
- Relationship between $\mathcal{F}_1$ and $\mathcal{F}_2$ (Bach, 2017):
  - $\mathcal{F}_2$ is included in $\mathcal{F}_1$. Moreover, for any $v \in \mathcal{V}$, $\varphi_v \in \mathcal{F}_1$ with $\gamma_1(\varphi_v) \leq 1$, while in general $\varphi_v \notin \mathcal{F}_2$.
  - $\mathcal{F}_1$ and $\mathcal{F}_2$ have very different properties (e.g., γ_2 may be computed easily in several cases, while γ_1 does not).
- $\mathcal{F}_2$ equipped with the norm γ_2 is a reproducing kernel Hilbert space (RKHS) with positive definite kernel $K(x, y) = \int_{\mathcal{V}} \varphi_v(x)\varphi_v(y) d\tau(v)$. (Bach, 2017)

Reproducing kernel Hilbert space

- Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a continuous function satisfying
  - $K(x, y) = K(y, x)$ for any $x, y \in \mathcal{X}$. (symmetric)
  - $\sum_{i,j=1}^{m} c_i c_j K(x_i, x_j) \geq 0$ for any $\{x_i\}_{i=1}^{m} \subset \mathcal{X}$, $\{c_i\}_{i=1}^{m} \subset \mathbb{R}$, and $m \in \mathbb{N}$. (positive semi-definite)
- Define $K_x : \mathcal{X} \to \mathbb{R}$ by $K_x(y) = K(x, y)$, for any $y \in \mathcal{X}$.
- Inner product: $\left\langle K_x, K_y \right\rangle_K = K(x, y)$, for any $x, y \in \mathcal{X}$.
- A reproducing kernel Hilbert space $\mathcal{H}_K$ is the completion of $\mathrm{Span}\{K_x, x \in \mathcal{X}\}$ completed w.r.t. $\langle \cdot, \cdot \rangle_K$.
- Reproducing property: $f(x) = \langle f, K_x \rangle_K$, for any $f \in \mathcal{H}_K, x \in \mathcal{X}$.

Supervised machine learning

- Let $\mathcal{X} \times \mathbb{R}$ be equipped with some distribution over the pairs $(x, y) \in \mathcal{X} \times \mathbb{R}$.
- Consider a loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.
- Our aim is to find a function $f : \mathcal{X} \to \mathbb{R}$ from a class $\mathcal{F}$ of functions equipped with a norm γ (e.g., $\mathcal{F}_1$ and $\mathcal{F}_2$ equipped with γ_1 and γ_2) such that the risk $\mathbb{E}_{(x,y)}[\ell(y, f(x))]$ is small.
- Given i.i.d. observations $\{(x_i, y_i)\}_{i=1}^n$, we consider the empirical risk minimization learning scheme to find a minimizer of the empirical risk $\frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i))$ over $\mathcal{F}$.
- Regularization:
  - Constraining f to be in a small ball $\mathcal{F}^\delta = \{f \in \mathcal{F}, \gamma(f) \le \delta\}$ with $\delta > 0$.
  - Regularizing the empirical risk by $\lambda\gamma(f)$ with $\lambda > 0$.

Random feature kernel

- Recall the RKHS $\mathcal{F}_2$ and the kernel function $K(x,x') = \int_{\mathcal{V}} \varphi_v(x)\varphi_v(x')d\tau(v)$.
- Let $\{v_i\}_{i=1}^m$ be a sample drawn independently from τ.
- We define the approximation

$$\hat{K}(x,x') = \frac{1}{m}\sum_{i=1}^{m} \varphi_{v_i}(x)\varphi_{v_i}(x'),$$

  which is a random feature representation (Rahimi and Recht, 2007).
- With a random feature kernel $\hat{K}$, one can do
  - Kernel ridge regression (Rudi and Rosasco, 2017).
  - Kernel-based stochastic gradient descent (SGD) learning algorithm (Carratino et al., 2018).

Kernel-based SGD

- Given a kernel function K and an i.i.d. sample $\{(x_i, y_i)\}_{i=1}^n$, consider a supervised learning problem with $f \in \mathcal{H}_K$.
- By the reproducing property, we rewrite

$$\ell(y_i, f(x_i)) = \ell\left(y_i, \left\langle f, K_{x_i} \right\rangle_K\right).$$

- Then the gradient of $\ell(y_i, f(x_i))$ w.r.t. f and $\langle \cdot, \cdot \rangle_K$ (kernel gradient) is given by $\ell'(y_i, f(x_i))K_{x_i} \in \mathcal{H}_K$, here ℓ' is the gradient of ℓ w.r.t. the second argument.
- The kernel-based SGD (for example, Kivinen et al., 2004) is defined iteratively by $f_0 = 0$ and

$$\begin{aligned} f_{k+1}(x) &= f_k(x) - \eta_k \ell'(y_k, f_k(x_k))K_{x_k}(x) \\ &= f_k(x) - \eta_k \ell'(y_k, f_k(x_k))K(x_k, x) \end{aligned}$$

with η_k being the step-size.

SGD updating parameters

- Consider a function $f(x;\theta)$ belonging to some class of functions parametrized by $\theta = (\theta_1, \cdots, \theta_P)^\top \in \mathbb{R}^P$ with P being the dimension of the parameter space.
- The parameter can be updated by SGD as follows

$$\theta^{k+1} = \theta^k - \eta_k \ell'(y_k, f(x_k;\theta^k)) \nabla f(x_k;\theta^k)$$

  with some initialization θ^0, where ∇ is the gradient w.r.t. θ.
- If, instead, we consider updating a function in each iteration, one (Chizat and Bach, 2018; Jacot et al., 2018) has the first order approximation

$$f(x;\theta^{k+1}) \approx f(x;\theta^k) - \eta_k \nabla f(x;\theta^k)^\top \nabla f(x_k;\theta^k) \ell'(y_k, f(x_k;\theta^k)),$$

  which is a kernel-based SGD algorithm with kernel $K_{\theta^k}(x,x') = \nabla f(x;\theta^k)^\top \nabla f(x';\theta^k)$.
- The kernel function depends on the parameter θ^k and we hope that θ^k remains in a neighborhood of θ^0 during the training process.

Some remarks

Recall the iteration

$$f(x;\theta^{k+1}) = f(x;\theta^k) - \eta_k \nabla f(x;\theta^k)^\top \nabla f(x_k;\theta^k)\ell'(y_k, f(x_k;\theta^k)),$$

which is referred to as lazy training (Chizat and Bach, 2018).

- (Chizat and Bach, 2018) The key point is that if the iterates remain in a neighborhood of θ^0 then this kernel is roughly constant throughout training. When $f(x;\theta^0) \approx 0$, this behavior naturally arises when scaling the model as αf with a large scaling factor $\alpha > 0$. Indeed, this scaling does not change the tangent model and brings the iterates of SGD closer to θ^0 by a factor $1/\alpha$.
- For the linear case $f(x;\theta) = \frac{1}{\sqrt{P}} \sum_{i=1}^{P} \theta_i \varphi_{v_i}(x)$ with given random features $\{\varphi_{v_i}\}_{i=1}^P$, we have $\nabla f(x;\theta) = \frac{1}{\sqrt{P}}(\varphi_{v_i}(x))_{i=1}^P$ and

$$K_\theta(x,x') = \nabla f(x;\theta)^\top \nabla f(x';\theta) = \frac{1}{P}\sum_{i=1}^{P} \varphi_{v_i}(x)\varphi_{v_i}(x'),$$

  which is a random feature kernel!

Linear approximation around initialization

- Given initial parameter $\theta^0 \in \mathbb{R}^P$, consider the linear approximation of $f(x;\theta)$ around θ^0,

$$T_f(x;\theta) = f(x;\theta^0) + (\theta - \theta^0)^\top \nabla f(x;\theta^0).$$

- The corresponding function class is affine in θ while, in general, is not affine in x.
- $T_f(x;\theta)$ is called the tangent model (Chizat and Bach, 2018).

Kernel method with an offset

- Consider a loss function $\ell(y,t)$ with $\ell'(y,t)$ depending only on $y-t$ such as the quadratic loss $\ell(y,t)=(y-t)^2$.
- We have

$$\begin{aligned}\nabla_\theta \ell(y,T_f(x;\theta)) &= \ell'(y,T_f(x;\theta))\nabla f(x;\theta^0)\\ &= \ell'(y-f(x;\theta^0),(\theta-\theta^0)^\top \nabla f(x;\theta^0))\nabla f(x;\theta^0)\\ &= \nabla_\theta \ell(y-f(x;\theta^0),(\theta-\theta^0)^\top \nabla f(x;\theta^0))\end{aligned}$$

- This is equivalent to a kernel method with the tangent kernel

$$K(x,x') = \nabla f(x;\theta^0)^\top \nabla f(x';\theta^0)$$

with the output variable y shifted by $f(x;\theta^0)$.

Neural tangent kernel, I

- Consider a single-hidden-layer no biases neural network

$$f(x;\theta) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} s_j \sigma(w_j^\top x), w_j \in \mathbb{R}^d, s_j \in \mathbb{R}$$

  with an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ and parameter $\theta = (w_1^\top, \cdots, w_m^\top, s_1, \cdots, s_m)^\top \in \mathbb{R}^{m(d+1)}$. $(P = m(d+1))$
- To set the function to 0 at initialization, one may consider networks of width $2m$ of the form

$$1^\top \sigma(W_+ x) - 1^\top \sigma(W\_x),$$

  where $W_\pm = W$ at initialization with $W \in \mathbb{R}^{m \times d}$ being a Gaussian matrix and σ acts on vectors componentwise.
- $$\nabla_{w_j} f(x;\theta) = \frac{s_j}{\sqrt{m}} \sigma'(w_j^\top x) x, \quad \nabla_{s_j} f(x;\theta) = \frac{1}{\sqrt{m}} \sigma(w_j^\top x)$$

Neural tangent kernel, II

- The neural tangent kernel (NTK) is then given by

$$\begin{aligned} K_m(x,x') &= \nabla f(x,\theta)^\top \nabla f(x',\theta) \\ &= \frac{1}{m}\sum_{j=1}^m (x^\top x') s_j^2 \sigma'(w_j^\top x)\sigma'(w_j^\top x') + \frac{1}{m}\sum_{j=1}^m \sigma(w_j^\top x)\sigma(w_j^\top x') \\ &=: K_m^{(1)}(x,x') + K_m^{(2)}(x,x'), \end{aligned}$$

  which is the sum of two random feature kernels $K_m^{(1)}$ and $K_m^{(2)}$.

- If the weights w_j (resp. s_j) are drawn independently from a distribution on $\mathbb{R}^d$ (resp. a distribution on $\mathbb{R}$), then $K_m^{(1)}$ and $K_m^{(2)}$ converges to

$$K^{(1)}(x,x') = (x^\top x')\mathbb{E}_{(s,w)}\left[s^2\sigma'(w^\top x)\sigma'(w^\top x')\right]$$

  and

$$K^{(2)}(x,x') = \mathbb{E}_w[\sigma(w^\top x)\sigma(w^\top x')],$$

  respectively, as $m \to +\infty$.

Closed form for ReLU

- Let $\sigma(t) = \max\{t, 0\}$ be the rectified linear unit (ReLU) activation.
- ReLU is not differentiable at 0. One may consider its subdifferential $[0, 1]$ at 0.
- When the activation function is ReLU and w is spherically symmetric (e.g., a standard Gaussian distribution), one has the following closed form (Cho and Saul, 2009):

$$K^{(1)}(x, x') = \frac{x^\top x' \mathbb{E}[s^2]}{2\pi}(\pi - \eta)$$

and

$$K^{(2)}(x, x') = \frac{\|x\|\|x'\|\mathbb{E}[\|w\|^2]}{2\pi d}((\pi - \eta)\cos\eta + \sin\eta),$$

where $\eta = \mathrm{acrcos}\left(\frac{x^\top x'}{\|x\|\|x'\|}\right)$ is the angle between x and x'.

Special case in the lecture notes (Telgarsky, 2021)

- Single hidden layer, no biases, train only layer 1:

$$f(x;W) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} s_j \sigma(w_j^\top x), x \in \mathbb{R}^d, w_j \in \mathbb{R}^d, s_j \in \{\pm 1\}.$$

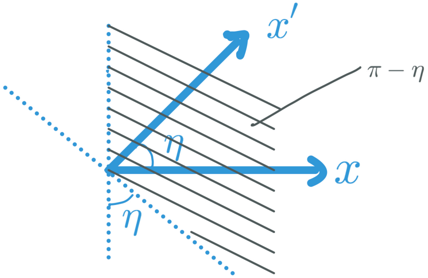
  Here s_j will not be trained and $W = (w_1, \cdots, w_m)^\top \in \mathbb{R}^{m \times d}$ is the parameter.
- Consider the following linear approximation around initialization $W_0 = (w_{1,0}, \cdots, w_{m,0})^\top \in \mathbb{R}^{m \times d}$.

$$\begin{aligned} W \mapsto & \frac{1}{\sqrt{m}} \sum_{j=1}^{m} s_j \left[\sigma(w_{j,0}^\top x) + \left(w_j - w_{j,0}\right)^\top x \sigma'(w_{j,0}^\top x)\right] \\ = & \frac{1}{\sqrt{m}} \sum_{j=1}^{m} s_j \left[\sigma(w_{j,0}^\top x) - w_{j,0}^\top x \sigma'(w_{j,0}^\top x)\right] + \frac{1}{\sqrt{m}} x^\top \sum_j s_j w_j \sigma'(w_{j,0}^\top x) \end{aligned}$$

- For ReLU activation, there holds $\sigma(t) = t\sigma'(t)$ and the first term disappears. The corresponding NTK is only $K_m^{(1)}$.

Proof of the closed form of $K^{(1)}$

- Since $s \in \{\pm 1\}$, we have $K^{(1)}(x, x') = x^\top x' \mathbb{E}_w \left(\mathbb{1}[w^\top x \geq 0] \mathbb{1}[w^\top x' \geq 0] \right)$. We need w to have nonnegative inner product with x and x', which corresponds only to the angle between w and x and the angle between w and x'.
- All that matters is the plane spanned by $\{x, x'\}$;
- Let $\eta = \arccos\left(\frac{x^\top x'}{\|x\|\|x'\|}\right)$ be the angle between x and x'. Since w is spherically symmetric, the probability of success is $\frac{\pi - \eta}{2\pi}$.



Linear Approximation around 0

- Consider the linear approximation (w.r.t. W) of $f(x;W)$ at 0:

$$W \mapsto \frac{1}{\sqrt{m}} \sum_j s_j \left(\sigma(0) + (w_j - 0)^\top x \sigma'(0)\right)$$

$$= \frac{\sigma(0)}{\sqrt{m}} \sum_j s_j + \frac{\sigma'(0)}{\sqrt{m}} x^\top \left(\sum_j s_j w_j\right)$$

- A linear predictor
  - This expression is affine in x.
  - Gradients of this w.r.t. different w_j are rescalings (by s_j) of each other.
- The corresponding tangent kernel is the linear kernel

$$K^{\mathrm{Lin}}(x, x') = x^\top x'.$$

Thank You!